Direct memory access

From Wikipedia, the free encyclopedia

Direct memory access (DMA) is a feature of modern computers, that allows certain hardware subsystems within the computer to access system memory for reading and/or writing independently of the central processing unit. Many hardware systems use DMA including disk drive controllers, graphics cards, network cards, and sound cards. Computers that have DMA channels can transfer data to and from devices with much less CPU overhead than computers without a DMA channel.

Without DMA, using programmed input/output (PIO) mode, the CPU typically has to be occupied for the entire time it's performing a transfer. With DMA, the CPU would initiate the transfer, do other operations while the transfer is in progress, and receive an interrupt from the DMA controller once the operation has been done. This is especially useful in real-time computing applications where not stalling behind concurrent operations is critical.

Contents

Principle

DMA is an essential feature of all modern computers, as it allows devices to transfer data without subjecting the CPU to a heavy overhead. Otherwise, the CPU would have to copy each piece of data from the source to the destination. This is typically slower than copying normal blocks of memory since access to I/O devices over a peripheral bus is generally slower than normal system RAM. During this time the CPU would be unavailable for any other tasks involving CPU bus access, although it could continue doing any work which did not require bus access.

A DMA transfer essentially copies a block of memory from one device to another. While the CPU initiates the transfer, it does not execute it. For so-called "third party" DMA, as is normally used with the ISA bus, the transfer is performed by a DMA controller which is typically part of the motherboard chipset. More advanced bus designs such as PCI typically use bus mastering DMA, where the device takes control of the bus and performs the transfer itself.

A typical usage of DMA is copying a block of memory from system RAM to or from a buffer on the device. Such an operation does not stall the processor, which as a result can be scheduled to perform other tasks. DMA transfers are essential to high performance embedded systems. It is also essential in providing so-called zero-copy implementations of peripheral device drivers as well as functionalities such as network packet routing, audio playback and streaming video.

DMA engines

In addition to hardware interaction, DMA can also be used to offload expensive memory operations, such as large copies or scatter-gather operations, from the CPU to a dedicated DMA engine. While normal memory copies are typically too small to be worthwhile to offload on today's desktop computers, they are frequently offloaded on embedded devices due to more limited resources.[1]

Newer Intel Xeon processors also include a DMA engine technology called I/OAT, meant to improve network performance on high-throughput network interfaces, such as gigabit Ethernet, in particular.[2] However, benchmarks with this approach on Linux indicate no more than 10% improvement in CPU utilization.[3]

Examples

ISA

For example, a PC's ISA DMA controller has 16 DMA channels of which 7 are available for use by the PC's CPU. Each DMA channel has associated with it a 16-bit address register and a 16-bit count register. To initiate a data transfer the device driver sets up the DMA channel's address and count registers together with the direction of the data transfer, read or write. It then instructs the DMA hardware to begin the transfer. When the transfer is complete, the device interrupts the CPU.

"Scatter-gather" DMA allows the transfer of data to and from multiple memory areas in a single DMA transaction. It is equivalent to the chaining together of multiple simple DMA requests. Again, the motivation is to off-load multiple input/output interrupt and data copy tasks from the CPU.

DRQ stands for DMA request; DACK for DMA acknowledge. These symbols are generally seen on hardware schematics of computer systems with DMA functionality. They represent electronic signaling lines between the CPU and DMA controller.

See also

- Remote Direct Memory Access

References

- mmap() and DMA (http://www.xml.com/ldd/chapter/book/ch13.html), from *Linux Device Drivers, 2nd Edition*, Alessandro Rubini & Jonathan Corbet
- Memory Mapping and DMA (http://www.oreilly.com/catalog/linuxdrive3/book/ch15.pdf), from *Linux Device Drivers, 3rd Edition*, Jonathan Corbet, Alessandro Rubini, Greg Kroah-Hartman
- DMA and Interrupt Handling (http://www.eventhelix.com/RealtimeMantra/FaultHandling/dma_interrupt_handling.htm)
- DMA Modes & Bus Mastering (http://www.pcguide.com/ref/hdd/if/ide/modesDMA-c.html)

1. ^ Ganssle, Jack (1994-10). "Memory copies in hardware (http://www.ganssle.com/articles/adma.htm)". *Embedded Systems Programming*. Retrieved on

2006-11-12.
2. ^ Corbet, Jonathan (2005-12-06). "Memory copies in hardware (http://lwn.net/Articles/162966/)". *LWN.net* (December 8, 2005). Retrieved on 2006-11-12.
3. ^ Grover, Andrew (2006-06-01). I/OAT on LinuxNet wiki (http://linux-net.osdl.org/index.php/I/OAT). *Overview of I/OAT on Linux, with links to several benchmarks.* Retrieved on 2006-12-12.

Retrieved from "http://en.wikipedia.org/wiki/Direct_memory_access"

Categories: Computer memory | Motherboard